(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112784458 A
(43)申请公布日 2021.05.11

(21)申请号 202110090133.2

(22)申请日 2021.01.22

(71)申请人 南京理工大学
地址 210094 江苏省南京市孝陵卫200号

(72)发明人 张鑫 周成 于存贵 邹利波
魏甫全 侯晓明

(74)专利代理机构 南京理工大学专利中心 32203
代理人 汪清

(51)Int.Cl.
G06F 30/23(2020.01)
G06T 17/00(2006.01)

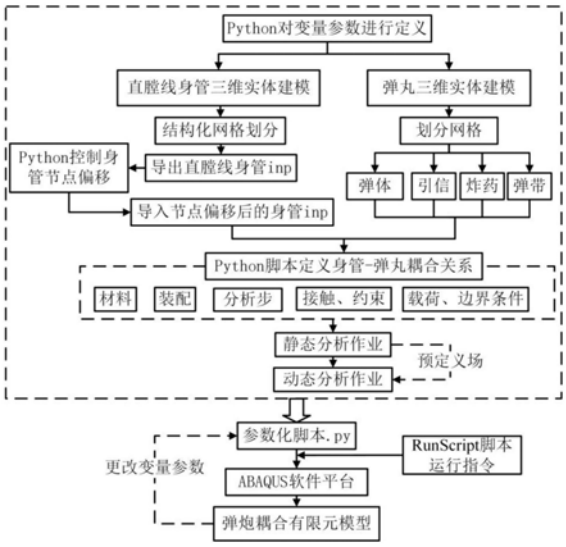
权利要求书2页 说明书8页 附图8页

(54)发明名称

一种弹炮耦合有限元参数化建模方法

(57)摘要

本发明提出了一种弹炮耦合有限元参数化建模方法,涉及机械结构参数化建模技术领域,将线膛身管弹炮耦合有限元建模过程通过Python语言进行描述,使用脚本建立弹炮耦合有限元模型,使得建立弹炮耦合有限元模型更加方便、快捷;主要包括:首先建立身管及弹丸二维草图,将二维草图实现全约束并定义参数化变量,根据二维草图自动生成三维模型,并根据缠角变化规律采用Python语言对身管节点进行偏移,采用Python描述建立弹丸与身管之间的耦合关系。本发明采用编程技术实现了弹炮耦合有限元模型的自动更改与数值模型的自动更新,从而提高了弹炮耦合有限元模型的建模速度,缩短了设计周期,极大提高了工作效率。



CN 112784458 A

1.一种弹炮耦合有限元参数化建模方法,包括以下步骤:

步骤1,根据身管实际结构尺寸,确定结构参数变量,利用Python语言定义结构参数化变量,编写有限元软件建模初始化程序;

步骤2,根据弹丸及身管实际结构,在有限元软件中建立直膛线身管及弹丸二维草图,进而建立三维实体模型,从建模过程中软件后台自动生成的rpy文件中提取出相应程序段,得到身管及弹丸三维实体建模程序;

步骤3,使用Python语言控制直膛线身管及弹丸的有限元单元网格的自动化划分;

步骤4,导出直膛线身管有限元inp文件,利用Python语言读取身管单元节点并进行单元节点偏移,生成具有螺旋膛线的身管有限元模型inp文件;

步骤5,利用Python语言读取步骤4生成的inp文件并将模型导入到有限元软件;

步骤6,通过在有限元软件ABAQUS中对身管及弹丸模型进行材料属性定义、装配定义、分析步定义、接触约束定义、载荷及边界条件加载操作,将操作过程中软件后台自动生成的rpy文件中的程序语句提取出来,得到身管和弹丸耦合关系施加的Python程序语句;

步骤7,在第6步装配体模型的基础上,编写静态与动态分析作业自动创建程序语句;

步骤8,将建模过程程序进行集成整合,建立弹炮耦合有限元模型脚本,通过脚本运行指令,建立弹炮耦合有限元模型。

2.根据权利要求1所述的弹炮耦合有限元参数化建模方法,其特征在于,所述步骤2中的直膛线身管炮口端面膛线草图采用全约束,且膛线边界线之间通过角度进行约束。

3.根据权利要求1所述的弹炮耦合有限元参数化建模方法,其特征在于,步骤4进行单元节点偏移,具体包括以下步骤:

4.1)定义身管装配体,在装配体模块定义身管装配体模型;

4.2)导出直膛线身管inp文件,将身管装配体模型的Job-SG.inp文件导出;

4.3)计算身管膛线缠角沿炮膛轴线的变化规律,渐速段缠角正切值变化规律为$y'_1(x)$,等齐段缠角正切值变化规律为$y'_2(x)$:

$$\begin{cases} y_1'(x)=\dfrac{\tan\alpha_1-\tan\alpha_0}{l}\left(x+\dfrac{l\tan\alpha_0}{\tan\alpha_1-\tan\alpha_0}\right)(0\le x\le l) \\ y_2'(x)=\tan\alpha_1\,(l<x\le L) \end{cases}$$

其中,x为膛线上某一点与阴线起始点之间的距离,膛线初始缠角为α_0,炮口缠角为α_1,膛线总长为L,渐速段长度为l;

4.4)按照缠角变化规律进行身管节点偏移。

4.根据权利要求3所述的弹炮耦合有限元参数化建模方法,其特征在于,线膛身管有限元单元节点偏移过程为:

Python程序首先从炮尾中心点开始读取身管单元层坐标和节点坐标值,当坐标值x_0小于阴线起点与药室底端面之间的距离h时,单元节点坐标不变,当单元层x_0坐标值大于h且小于h+l时,根据缠角变化规律公式得到身管渐速段单元节点偏移后的坐标(x_0,y',z');当单元层x_0坐标值大于h+l时,进行等齐段身管节点的偏移,得到等齐段身管单元节点偏移后的坐标(x_0,y'',z''),随着单元识别层编号i的增加,单元节点按照缠角变化规律沿着身管轴线方向依次进行偏移,直到单元层x_0坐标值大于身管长度L_s时,身管单元节点偏移结束,得

到节点偏移后的混合膛线身管有限元模型Job-SG_NEW.inp文件。

5.根据权利要求1所述的弹炮耦合有限元参数化建模方法，其特征在于，步骤8建立弹炮耦合有限元模型，具体过程为：首先设置好变量的初始值，然后打开ABAQUS初始化界面，点开RunScript脚本运行指令，选中弹炮耦合参数化脚本程序py并点击运行，ABAQUS软件将按照程序代码，依次完成建立实体模型、划分网格、定义并赋予材料属性、设置接触与约束关系、施加载荷与边界条件。

6.根据权利要求1所述的弹炮耦合有限元参数化建模方法，其特征在于，所述装配后的弹丸与身管之间设置接触属性，接触行为采用罚函数法进行描述。

7.根据权利要求1所述的弹炮耦合有限元参数化建模方法，其特征在于，所述身管有限元单元节点根据膛线缠角变化规律进行偏移。

8.根据权利要求1所述的弹炮耦合有限元参数化建模方法，其特征在于，所述静态与动态分析作业之间通过预定义场进行联系与信息导入。

一种弹炮耦合有限元参数化建模方法

技术领域

[0001] 本发明属于结构参数化建模技术领域,特别是一种弹炮耦合有限元参数化建模技术领域。

背景技术

[0002] 弹丸发射是一个极其复杂的动力学过程。弹丸在膛内运动期间与炮膛之间相互作用,弹带材料在高温高压下发生高度非线性的力学行为,为了研究弹丸膛内时期的动力学规律,需要建立弹丸与身管相互作用的耦合模型。采用有限元法建立弹炮耦合模型对火炮进行发射动力学仿真分析是研究火炮系统射击过程内在规律的一种重要手段。传统建模方法非常耗时,且建模精度较低,为了能够深入研究弹炮相互作用的力学规律,往往需要建立大量的仿真模型进行动力学仿真分析。目前的建模方法难以保证快速建立出较高精度的弹炮耦合有限元模型,研究建立具有较高精度的参数化有限元模型对火炮发射动力学仿真具有重要意义。

[0003] 由于线膛身管内膛参数众多,弹丸与身管耦合作用极其复杂,加之弹丸与身管之间复杂的力学行为容易导致模型计算不收敛,这使得建立出具有较高精度的弹炮耦合有限元模型变得十分困难。现有技术解决弹炮耦合有限元建模的手段大多都是先在三维建模软件中分别建立身管及弹丸三维实体模型,然后将模型导入网格前处理软件中进行网格的划分,最后将划分好的身管网格模型导入有限元软件中,最后进行弹丸与身管的耦合关系的施加,从而建立出弹炮耦合有限元模型。采用以上建模方法存在的缺点在于:1)由于软件兼容性问题,三维实体模型导入网格划分软件后会经常发生精度问题,导致有限元网格划分失败。2)由于弹丸与身管模型耦合关系需要进行手动逐步操作,导致建模过程操作步骤繁琐、耗时。3)对于不同结构参数的弹炮耦合有限元模型,每当结构参数改变时,都需重新进行所有建模步骤的操作,导致建模过程自动化程度极低。

[0004] 建立线膛身管的精确有限元模型是研究弹炮耦合问题的关键。现有技术中,能够较好地解决复杂空间曲线特征有限元建模的方法为节点偏移法,如以公布的发明专利CN107679341A一种身管结构有限元参数化建模方法即采用节点偏移法对线膛身管进行参数化建模,但该方法只能建立身管的有限元参数化模型,未能解决弹丸与身管耦合的参数化建模问题。

发明内容

[0005] 本发明的目的在于提供一种弹炮耦合有限元参数化建模方法,以解决传统弹炮耦合有限元建模效率低的问题。通过有限元软件二次开发可以实现弹炮耦合有限元复杂模型的自动、快速、高效建立,无需借助第三方软件,只需在ABAQUS软件中运行脚本程序,即可实现弹炮耦合有限元模型的自动化建立。

[0006] 实现本发明目的的技术解决方案为:

[0007] 步骤1,根据身管实际结构尺寸,确定结构参数变量,利用Python语言定义结构参

数化变量,编写有限元软件建模初始化程序;

[0008] 步骤2,根据弹丸及身管实际结构,在有限元软件中建立直膛线身管及弹丸二维草图,进而建立三维实体模型,从建模过程中软件后台自动生成的rpy文件中提取出相应程序段,得到身管及弹丸三维实体建模程序;

[0009] 步骤3,使用Python语言控制直膛线身管及弹丸的有限元单元网格的自动化划分;

[0010] 步骤4,导出直膛线身管有限元inp文件,利用Python语言读取身管单元节点并进行单元节点偏移,生成具有螺旋膛线的身管有限元模型inp文件;

[0011] 步骤5,利用Python语言读取步骤4生成的inp文件并将模型导入到有限元软件;

[0012] 步骤6,通过在有限元软件ABAQUS中对身管及弹丸模型进行材料属性定义、装配定义、分析步定义、接触约束定义、载荷及边界条件加载操作,将操作过程中软件后台自动生成的rpy文件中的程序语句提取出来,得到身管和弹丸耦合关系施加的Python程序语句;

[0013] 步骤7,在第6步装配体模型的基础上,编写静态与动态分析作业自动创建程序语句;

[0014] 步骤8,将建模过程程序进行集成整合,建立弹炮耦合有限元模型脚本,通过脚本运行指令,建立弹炮耦合有限元模型。

[0015] 本发明与现有技术相比,其显著优点是:

[0016] (1)本发明极大提高了弹炮耦合有限元的建模的效率,将弹丸与身管有限元耦合建模过程采用程序语言编写出来,相关人员在建立弹炮耦合有限元模型时,只需将程序化脚本在有限元软件ABAQUS中运行后,即可快速建立出有限元模型,极大简化了建模过程,使得模型生成速度非常快速。

[0017] 本发明实现了身管内膛结构参数的参数化设置,对于不同结构参数的建模,只需更改参数化变量值,即可快速建立出有限元模型,同时与手动逐步操作建模过程相比较,由于建模过程使用程序化操作流程,从而降低了有限元模型出错的概率。

[0018] (2)本发明提高了弹炮耦合有限元模型的精度,建模过程由于避免了不同软件之间的交互转换,从而使得有限元模型的精度大大提高。

[0019] (3)本发明建立模型的种类广泛,该方法不仅适用于混合膛线身管弹炮耦合建模,对于渐速膛线和等齐膛线身管弹炮耦合建模同样适用。

附图说明

[0020] 图1是本发明的参数化建模方法具体实施操作流程。

[0021] 图2为身管内膛结构示意图(单位:mm)。

[0022] 图3为身管外形结构示意图(单位:mm)。

[0023] 图4为膛线横剖面结构示意图(单位:mm)。

[0024] 图5为膛线展开曲线示意图。

[0025] 图6为膛线截面二维草图(单位:mm)。

[0026] 图7为直筒圆柱身管有限元模型示意图。

[0027] 图8为弹丸结构示意图。

[0028] 图9是弹丸有限元网格模型示意图。

[0029] 图10为身管节点偏移程序逻辑图。

[0030] 图11是节点偏移后的线膛身管有限元模型示意图。

[0031] 图12为膛线部分有限元网格模型示意图。

[0032] 图13为有限元软件ABAQUS初始化界面。

[0033] 图14为有限元软件ABAQUS脚本运行界面。

[0034] 图15为弹丸与身管耦合的有限元模型示意图。

具体实施方式

[0035] 下面结合说明书附图对本发明做进一步的介绍。

[0036] 结合图1至图15对本发明内容进行具体介绍,运用该参数化建模方法建立弹炮耦合有限元模型,具体操作步骤如下:

[0037] 步骤1,Python语言定义参数化变量

[0038] 首先对建模初始化程序进行定义,语句如下:

[0039] from abaqus import*

[0040] from abaqusConstants import*

[0041] from caeModules import*

[0042] executeOnCaeStartup()

[0043] from abaqus import为定义软件为ABAQUS的语句,from abaqusConstants import语句含义为导入ABAQUS中的常量,from caeModules import语句含义为导入ABAQUS的建模模块,executeOnCaeStartup语句含义为对建模模块进行初始化运行。

[0044] 身管内膛结构如图2所示,其中,d为身管口径,L_s为身管总长;d_1为阴线直径,l_1为坡膛长度;d_2为药室圆柱部直径,l_2为药室圆柱部长度;d_3为药室连接锥大端直径,l_3为药室连接锥长度;d_4为药室底部直径,l_4为药室本体长度;h为阴线起点与药室底端面之间的距离;K为坡膛锥度。此外将膛线条数记为n。

[0045] 身管外形结构如图3所示,其中,d_5为身管炮口外圆直径,d_6炮口制退器安装台阶外圆直径,d_7为身管中部外圆直径,d_8为身管尾部前端直径,d_9为身管尾部外圆直径,L_5为药室底部端面与炮口制退器安装台阶端面之间的距离,L_6为药室底部端面与身管中部外圆端面之间的距离,L_7为身管尾部长度,L_s为身管总长。

[0046] 身管膛线横剖面图如图4所示,其中,a为阳线宽,b为阴线宽,t为膛线深。

[0047] $K=(d_2-d)/l_1$

[0048] $$h=l_1+l_2+l_3+l_4-\frac{t}{K/2}$$

[0049] 身管膛线曲线展开曲线如图5所示,混合膛线曲线由渐速段$l_{M_0M_1}$和等齐段$l_{M_1M_2}$组成,渐速段曲线方程为二次抛物线方程。

[0050] $$\begin{cases} y_1(x)=\dfrac{\tan\alpha_1-\tan\alpha_0}{2l}\left(x+\dfrac{l\tan\alpha_0}{\tan\alpha_1-\tan\alpha_0}\right)^2-\dfrac{l\tan^2\alpha_0}{2(\tan\alpha_1-\tan\alpha_0)}\left(0\le x\le l\right) \\ y_2(x)=\tan\alpha_1\cdot x-\dfrac{l(\tan\alpha_1-\tan\alpha_0)}{2}\left(l\le x\le L\right) \end{cases}$$

[0051] 其中,$y_1(x)$为渐速段曲线方程,$y_2(x)$为等齐段曲线方程,x为膛线上某一点与阴线

起始点之间的距离，膛线初始缠角为α_0，炮口缠角为α_1，膛线总长为L，渐速段长度为l。

[0052] 将α_0，α_1，l，t，a定义为结构参数化变量，并赋予初值。(α_0、α_1单位为：度。l、t、a单位为：mm)

[0053] α_0＝2.901，α_1＝7.778，l＝4518，t＝2.3，a＝3.8

[0054] 将身管内膛尺寸参数中的d、d_2、d_3、d_4、l_1、l_2、l_3、l_4、n和身管外形尺寸参数d_5、d_6、d_7、d_8、d_9、L_5、L_6、L_7、L_s均设置为结构参数常量。

[0055] 阴线宽度b和阴线直径d_1与结构参数化变量及结构参数常量之间存在以下关系：

[0056] 阴线宽度：$b=\frac{\pi d}{n}-a$，b/2初值为3.17，膛线条数n＝48，阳线半径为d/2＝77.47；

[0057] 阴线直径：d_1＝2t+d，d_1/2初值为79.77。

[0058] 步骤2，建立身管及弹丸三维实体模型。

[0059] 根据身管实际结构在ABAQUS软件中按照图2和图3建立直筒身管三维实体模型。如图6所示，按照图4在建立的身管直筒模型炮口端面绘制膛线截面草图，膛线条数为48条，膛线分界线之间通过角度进行约束，角度值为7.5度；对阳线和阴线通过半径尺寸进行约束，阴线半径初始值为79.77，阳线半径为77.47；炮口端面草图绘制完成后，采用拉伸切除命令得到身管直膛线轮廓。将身管建模过程中生成的rpy中相应的程序段提取出，并将程序段中的数值采用上述定义的五个参数化变量表示，得到直膛线身管实体建模脚本程序。

[0060] 弹丸结构示意图如图8所示，弹丸为底凹弹，弹丸由弹体、炸药、引信组成。根据弹丸的实际尺寸建立实体模型，提取出建模过程中rpy文件中相应的程序段，将程序段中的相关数值采用上述五个参数化变量表示，得到弹丸实体建模脚本程序。

[0061] 步骤3，自动划分实体模型网格

[0062] 采用结构化网格划分方法对直膛线身管实体模型进行网格离散，建立的直膛线身管网格模型如图7所示。弹丸有限元网格模型如图9所示。其中身管网格自动划分程序如下所示：

[0063] 设置身管网格种子大小

[0064] p＝mdb.models['Model-1'].parts['SG']

[0065] p.seedPart(size＝5,deviationFactor＝0.1,minSizeFactor＝0.1)

[0066] 设置网格单元类型为C3D8R

[0067] elemType＝mesh.ElemType(elemCode＝C3D8R,elemLibrary＝STANDARD,kinematicSplit＝AVERAGE_STRAIN,secondOrderAccuracy＝OFF,hourglassControl＝DEFAULT,distortionControl＝DEFAULT)

[0068] c＝p.cells

[0069] pickedRegions＝(cells,)

[0070] p.setElementType(regions＝pickedRegions,elemTypes＝(elemType,))

[0071] 步骤4，导出身管有限元模型inp文件，Python语言读取单元节点并按照膛线缠角变化规律进行偏移。

[0072] 规定炮尾指向炮口方向为x正方向，竖直向上为y正方向，z轴正向由右手法则确定。

[0073] 4.1)定义身管装配体，在装配体模块定义身管装配体模型。

[0074] a＝mdb.models['Model-1'].rootAssembly##定义装配体模块

[0075] a.DatumCsysByDefault(CARTESIAN)##定义身管坐标系为全局坐标系

[0076] p＝mdb.models['Model-1'].parts['SG']##定义身管装配体模型为Model-1

[0077] a.Instance(name＝'SG-1',part＝p,dependent＝ON)

[0078] 4.2)导出直膛线身管inp文件,将身管装配体模型Model-1的Job-SG.inp文件导出。

[0079] mdb.Job(name＝'Job-SG',model＝'Model-1',description＝″,type＝ANALYSIS,)

[0080] mdb.jobs['Job-SG'].writeInput(consistencyChecking＝OFF)

[0081] 4.3)计算身管膛线缠角沿炮膛轴线的变化规律,渐速段缠角正切值变化规律为$y'_1(x)$,等齐段缠角正切值变化规律为$y'_2(x)$:

[0082] $$\begin{cases} y_1'(x)=\dfrac{\tan\alpha_1-\tan\alpha_0}{l}\left(x+\dfrac{l\tan\alpha_0}{\tan\alpha_1-\tan\alpha_0}\right)(0\le x\le l) \\ y_2'(x)=\tan\alpha_1(l<x\le L) \end{cases}$$

[0083] 身管网格单元某一节点绕炮膛转过的角度θ(x)与节点距阴线起始点之间的距离x存在如下关系:

[0084] $$\theta(x)=\frac{2\pi}{\eta d}\cdot x=\frac{2\tan\alpha}{d}\cdot x$$

[0085] 其中,α为身管轴线方向某一单元处的膛线缠角,η为膛线的缠度。

[0086] 单元节点绕炮膛轴线偏移的角度:$$\begin{cases} \theta_1(x)=\dfrac{2y_1'(x)}{d}\cdot x(0\le x\le l) \\ \theta_2(x)=\dfrac{2y_2'(x)}{d}\cdot x(l<x\le L) \end{cases}$$

[0087] 其中,x为身管单元节点与阴线起始点之间的距离,$\theta_1(x)$为渐速段身管节点偏移规律,$\theta_2(x)$为等齐段身管节点偏移规律。

[0088] 原节点坐标偏移一定角度后的新节点坐标:

[0089] 渐速段身管节点坐标(y,z)偏移θ_1角度后的节点坐标(y′,z′):

$$\begin{cases} y'=y\cos\theta_1-z\sin\theta_1 \\ z'=y\sin\theta_1+z\cos\theta_1 \end{cases}$$

[0090] 其中,θ_1为身管某一节点在渐速段偏移的角度。

[0091] 等齐段身管节点坐标(y,z)偏移θ_2角度后的节点坐标(y″,z″):

$$\begin{cases} y''=y\cos\theta_2-z\sin\theta_2 \\ z''=y\sin\theta_2+z\cos\theta_2 \end{cases}$$

[0092] 其中,θ_2为身管某一节点在等齐段偏移的角度。

[0093] 4.4)按照缠角变化规律进行身管节点偏移

[0094] 如图10所示,线膛身管有限元单元节点偏移过程为:Python程序首先从炮尾中心点开始读取身管单元层坐标和节点坐标值,当坐标值x_0小于阴线起点与药室底端面之间的

距离h时，单元节点坐标不变，当单元层x_0坐标值大于h且小于h+l时，根据膛线缠角变化规律可得到身管渐速段单元节点偏移后的坐标(x_0,y′,z′)，同理，当单元层x_0坐标值大于h+l时，进行等齐段身管节点的偏移，得到等齐段节点偏移后的坐标(x_0,y″,z″)，随着单元识别层编号i的增加，单元节点按照缠角变化规律沿着身管轴线方向依次进行偏移，直到单元层x_0坐标值大于身管长度L_s时，身管单元节点偏移结束，得到节点偏移后的混合膛线身管有限元模型Job-SG_NEW.inp文件，其中Job-SG_NEW.inp文件中的新模型为Job-SG_NEW。

[0095] 进一步地，作为一种实施方式，当l＝L时，可以进行渐速膛线身管的有限元建模。

[0096] 进一步地，作为另外一种实施方式，当l＝0时，可以进行等齐膛线身管的有限元建模。

[0097] 步骤5，将单元节点偏移后的身管Job-SG_NEW.inp文件通过Python语言读取并导入到ABAQUS中，即得到螺旋膛线身管有限元网格模型。

[0098] 5.1)导入节点偏移后的身管网格模型，相应的导入程序语句如下所示：

[0099] mdb.ModelFromInputFile(name＝'Job-SG_NEW',inputFileName＝'Job-SG_NEW.inp')

[0100] mdb.ModelFromInputFile目的在于读取模型Job-SG_NEW

[0101] 5.2)导入部件及装配体属性

[0102] import part##导入部件属性

[0103] import assembly##导入装配体属性

[0104] ##将模型Job-SG_NEW中的螺旋身管部件PART-SG导入到模型Model-1中

[0105] mdb.models['Model-1'].Part('PART-SG',mdb.models['Job-SG_NEW'].parts['PART-SG'])

[0106] ##定义身管装配体部件PART-SG-1

[0107] a＝mdb.models['Model-1'].rootAssembly

[0108] mdb.models['Model-1'].Instance('PART-SG-1',mdb.models['Job-SG_NEW'].rootAssembly.instances['PART-SG-1'])

[0109] 节点偏移后的身管有限元模型如图11所示，膛线结构如图12所示。

[0110] 步骤6，定义弹丸及身管的耦合关系。通过在有限元软件ABAQUS中对身管及弹丸模型进行材料属性定义、装配定义、分析步定义、接触约束定义、载荷及边界条件加载操作。引信、炸药、弹带分别与弹丸本体绑定约束。定义接触属性，弹带、前定心部与身管内膛之间设置接触属性，切向摩擦系数取0.1，接触属性采用罚函数法描述。模型采用0号装药，弹底施加由内弹道方程组解算出的压力曲线。边界条件为身管尾部全约束。将操作过程中软件后台自动生成的rpy文件中程序语句提取出来，得到身管、弹丸耦合关系施加的Python程序语句。

[0111] 其中创建有限元显式动态分析步的相应程序段如下所示：

[0112] mdb.models['Model-1'].ExplicitDynamicsStep(name＝'Step-1',previous＝'Initial',timePeriod＝0.016,improvedDtMethod＝ON)

[0113] 步骤7，创建静态、动态分析作业。

[0114] 复制模型并创建静态分析步

[0115] mdb.Model(name＝'Model-2',objectToCopy＝mdb.models['Model-1'])

[0116] mdb.models['Model-2'].StaticStep(name＝'Step-1',previous＝'Initial',maintainAttributes＝True,nlgeom＝ON)

[0117] 其中,Model-1为动态分析模型,Model-2为静态分析模型。

[0118] 7.1)创建静态分析作业

[0119] mdb.Job(name＝'Job-2',model＝'Model-2',description＝″,type＝ANALYSIS,atTime＝None,waitMinutes＝0,waitHours＝0,queue＝None,memory＝90,memoryUnits＝PERCENTAGE,getMemoryFromAnalysis＝True,explicitPrecision＝SINGLE,nodalOutputPrecision＝SINGLE,echoPrint＝OFF,modelPrint＝OFF,contactPrint＝OFF,historyPrint＝OFF,userSubroutine＝″,scratch＝″,resultsFormat＝ODB,multiprocessingMode＝DEFAULT,numCpus＝4,numDomains＝4,numGPUs＝0)

[0120] 7.2)创建动态分析作业

[0121] mdb.jobs['Job-2'].waitForCompletion()暂停当前脚本执行,直到当前计算任务终止

[0122] mdb.Job(name＝'Job-1',model＝'Model-1',description＝″,type＝ANALYSIS,atTime＝None,waitMinutes＝0,waitHours＝0,queue＝None,memory＝90,memoryUnits＝PERCENTAGE,explicitPrecision＝DOUBLE_PLUS_PACK,nodalOutputPrecision＝FULL,echoPrint＝OFF,modelPrint＝OFF,contactPrint＝OFF,historyPrint＝OFF,userSubroutine＝″,scratch＝″,resultsFormat＝ODB,parallelizationMethodExplicit＝DOMAIN,numDomains＝4,activateLoadBalancing＝False,multiprocessingMode＝DEFAULT,numCpus＝4)

[0123] 7.3)进行重启动设置,设置预定义场,实现静态分析作业与动态分析作业信息的交互。利用隐式求解器计算重力场作用下的初始发射状态,设置重启动分析,将重力场计算结果作为初始条件施加在动态显式分析步中。

[0124] 静态和动态分析步之间设置预定义场,相关程序段如下所示:

[0125] mdb.models['Model-1'].InitialState(updateReferenceConfiguration＝ON,fileName＝'Model-1',endStep＝LAST_STEP,endIncrement＝STEP_END,name＝'Predefined Field-1',createStepName＝'Initial',instances＝instances)

[0126] 其中,endStep＝LAST_STEP含义为将静态分析步的最后一个分析步计算出的信息导入到动态分析步中,预定义场被命名为Predefined Field-1,分析步命名为Initial。

[0127] 7.4)保存模型

[0128] mdb.saveAs(pathName＝'1')

[0129] 步骤8,将步骤1-步骤7中的Python脚本程序进行整合后,得到弹炮耦合参数化脚本程序。采用ABAQUS二次开发技术,将弹炮耦合有限元建模过程通过Python脚本程序编写出来,实现弹炮耦合有限元模型的快速建立。建立以初始缠角α_0、炮口缠角α_1、渐速段长度l、膛线深t和阳线宽a为变量参数的弹炮耦合有限元模型时,如图13所示,首先设置好变量的初始值,然后打开ABAQUS初始化界面,点开RunScript脚本运行指令,如图14所示,选中弹炮耦合参数化脚本程序py并点击运行,ABAQUS软件将按照程序代码,依次完成建立实体模型、划分网格、定义并赋予材料属性、设置接触与约束关系、施加载荷与边界条件。整个建模过程全部交由脚本程序一键完成。建立的弹炮耦合有限元模型如图15所示。

[0130] 以某大口径火炮身管内膛实际结构参数建立的弹炮耦合有限元模型，本次建模过程所花费时间不到一分钟，与传统建模方法相比，运用该参数化脚本程序建立弹炮耦合模型的效率会极大提高。

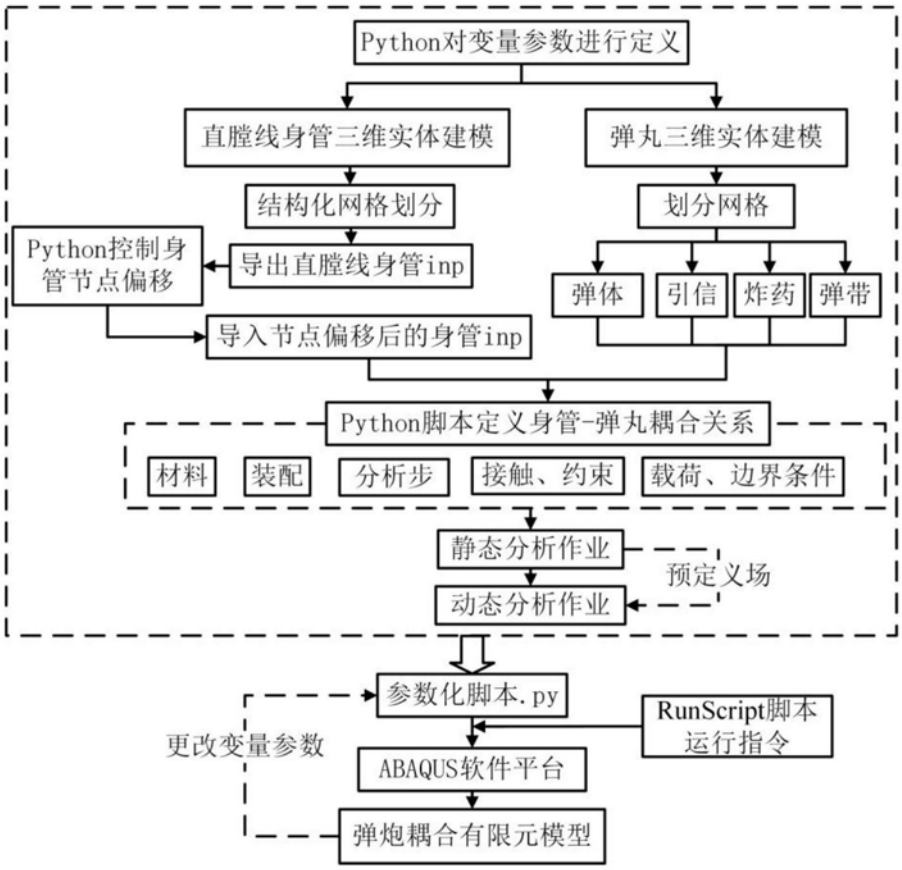

Python对变量参数进行定义
直膛线身管三维实体建模
弹丸三维实体建模
结构化网格划分
划分网格
Python控制身管节点偏移
导出直膛线身管inp
弹体
引信
炸药
弹带
导入节点偏移后的身管inp
Python脚本定义身管-弹丸耦合关系
材料
装配
分析步
接触、约束
载荷、边界条件
静态分析作业
预定义场
动态分析作业
参数化脚本.py
RunScript脚本运行指令
更改变量参数
ABAQUS软件平台
弹炮耦合有限元模型

图1

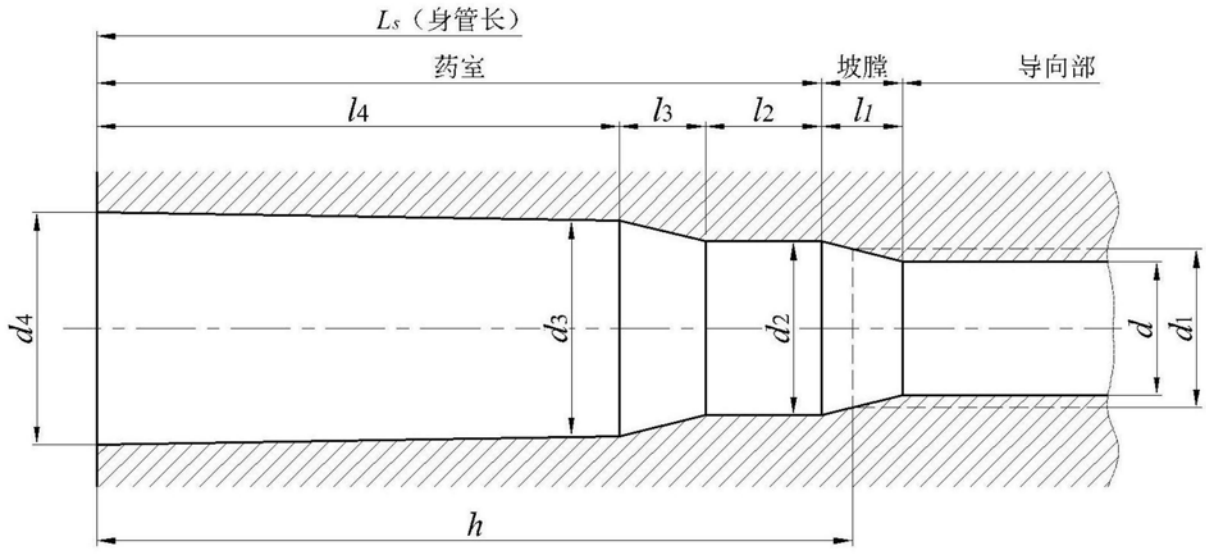

Ls（身管长）
药室
坡膛
导向部
l4
l3
l2
l1
d4
d3
d2
d
d1
h

图2

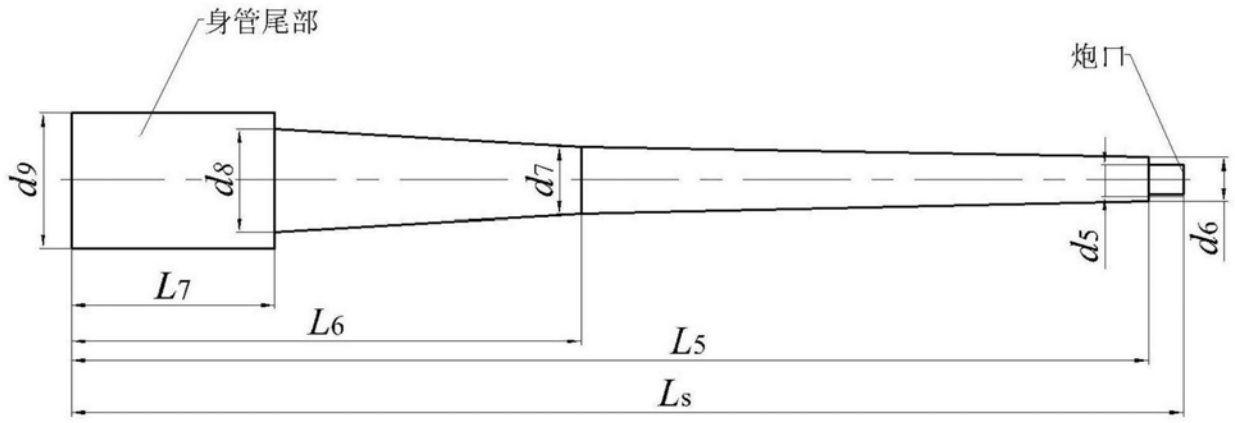


图3

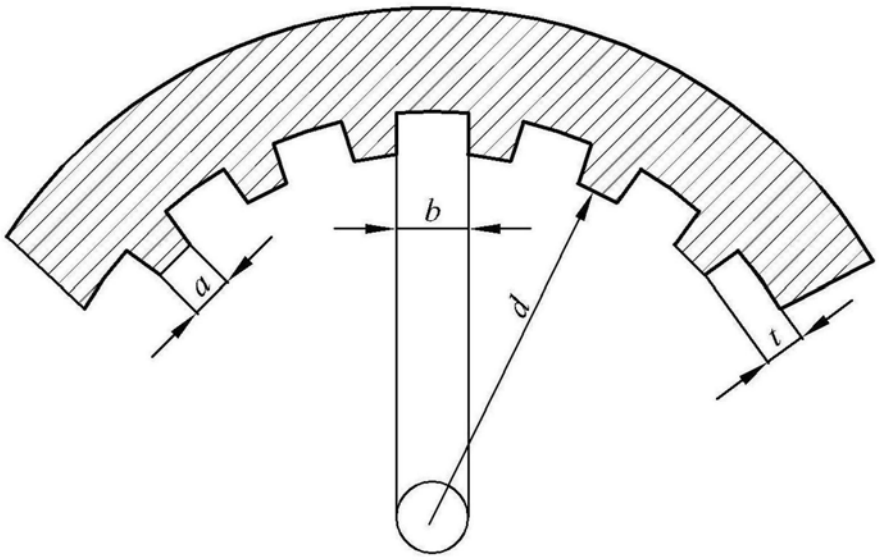


图4

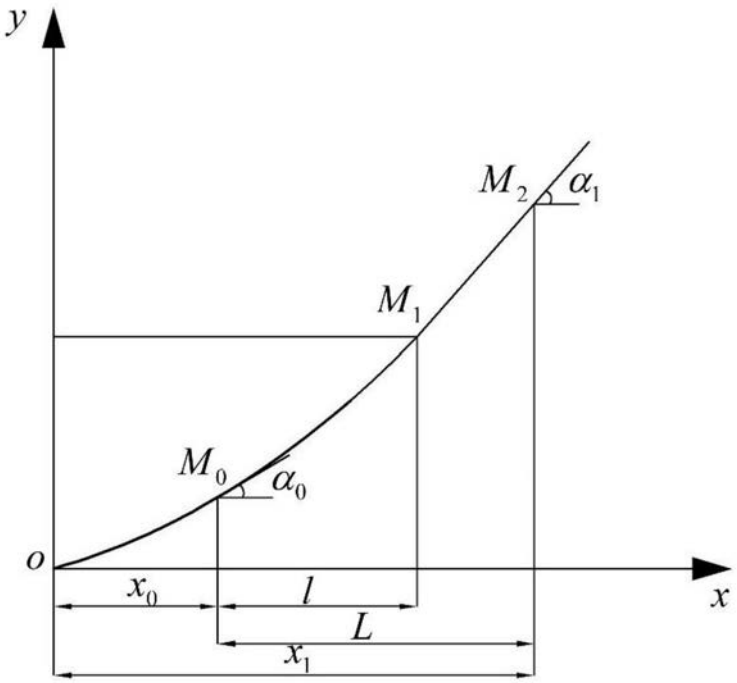


图5

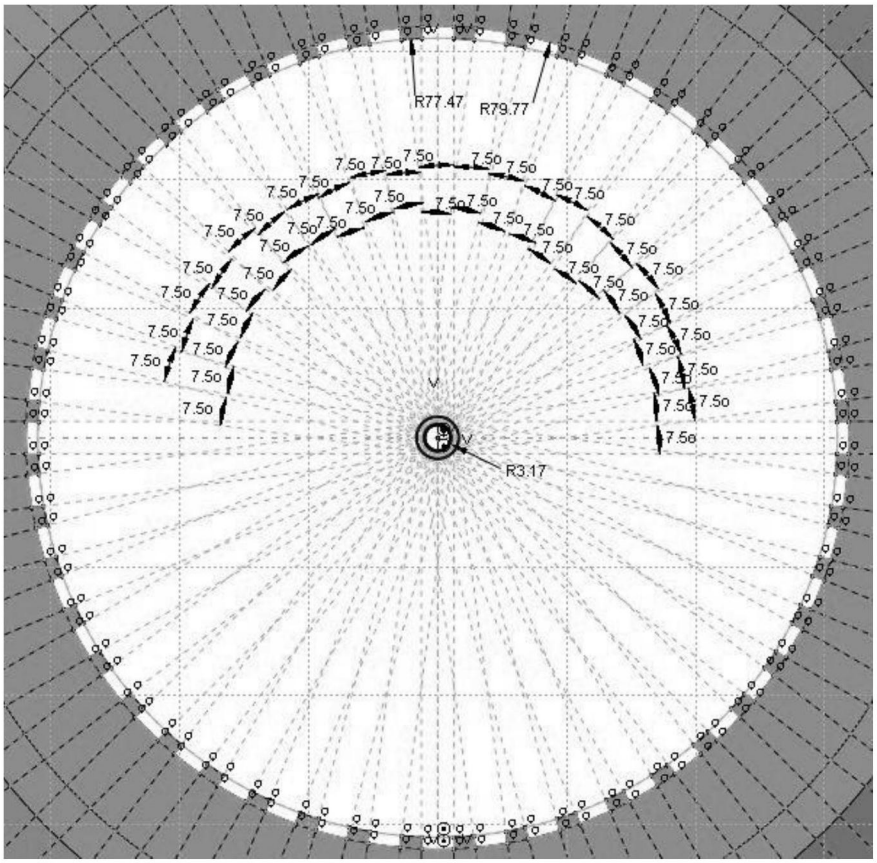


图6

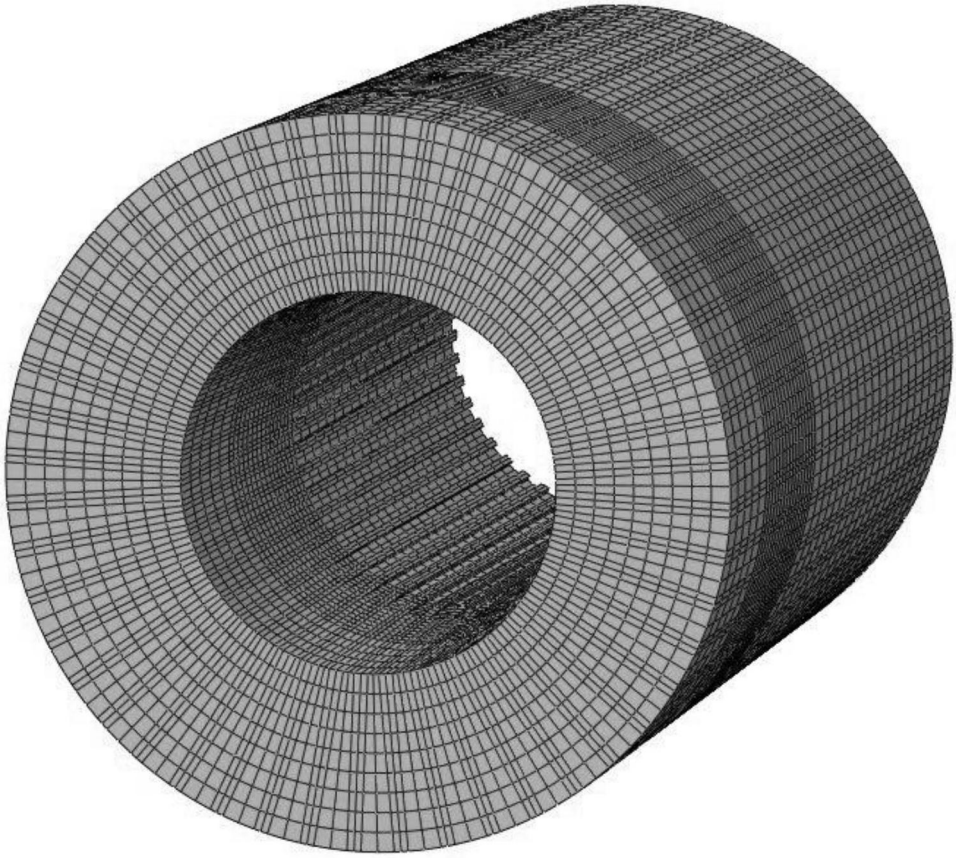


图7

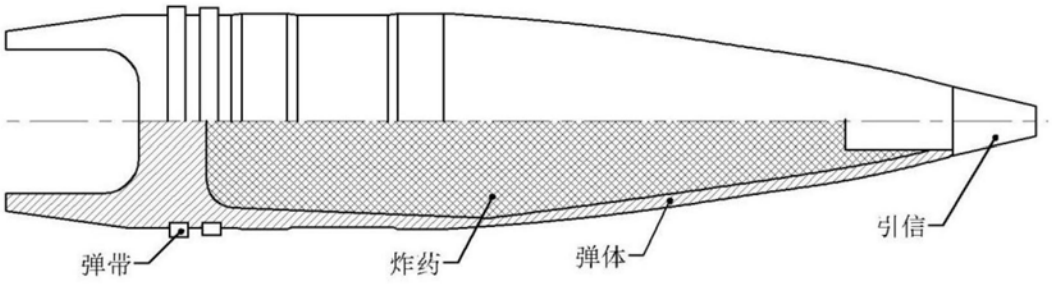


图8

图9

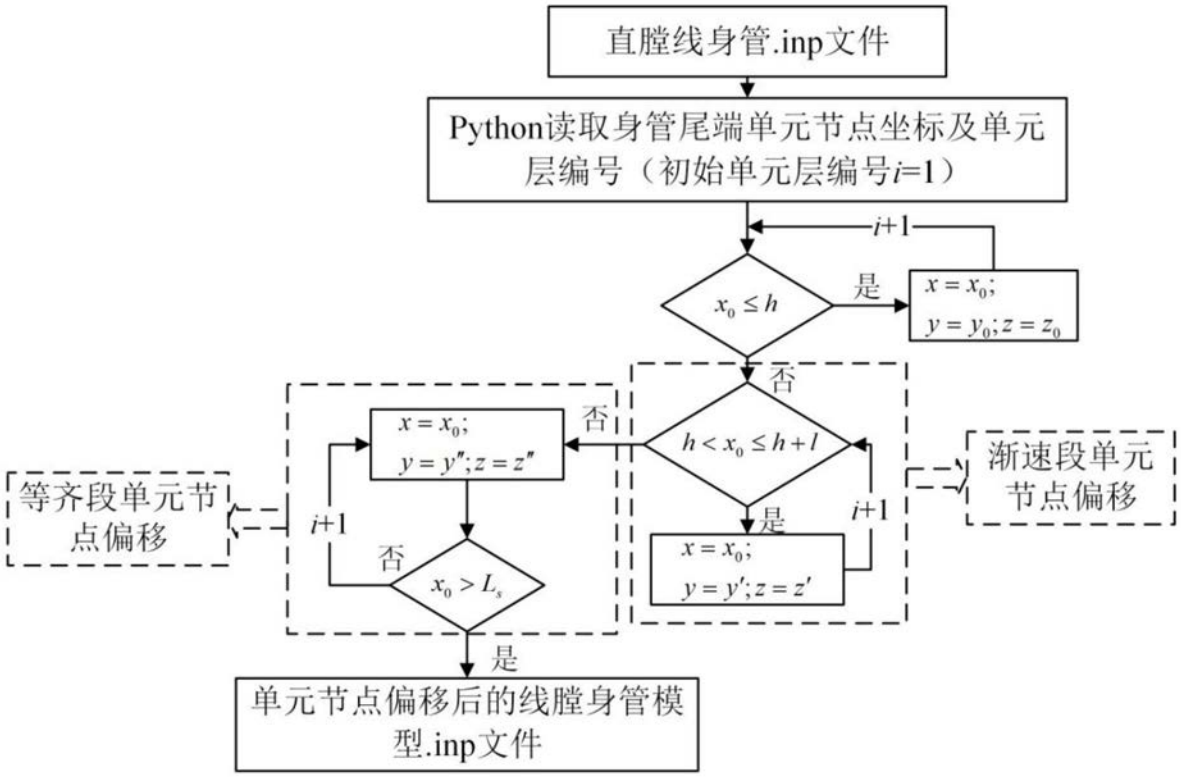

直膛线身管.inp文件
Python读取身管尾端单元节点坐标及单元层编号（初始单元层编号i=1）
i+1
$x_0 \le h$
是
$x = x_0;$
$y = y_0; z = z_0$
否
$h < x_0 \le h + l$
否
$x = x_0;$
$y = y''; z = z''$
等齐段单元节点偏移
i+1
否
$x_0 > L_s$
是
i+1
$x = x_0;$
$y = y'; z = z'$
是
渐速段单元节点偏移
单元节点偏移后的线膛身管模型.inp文件

图10

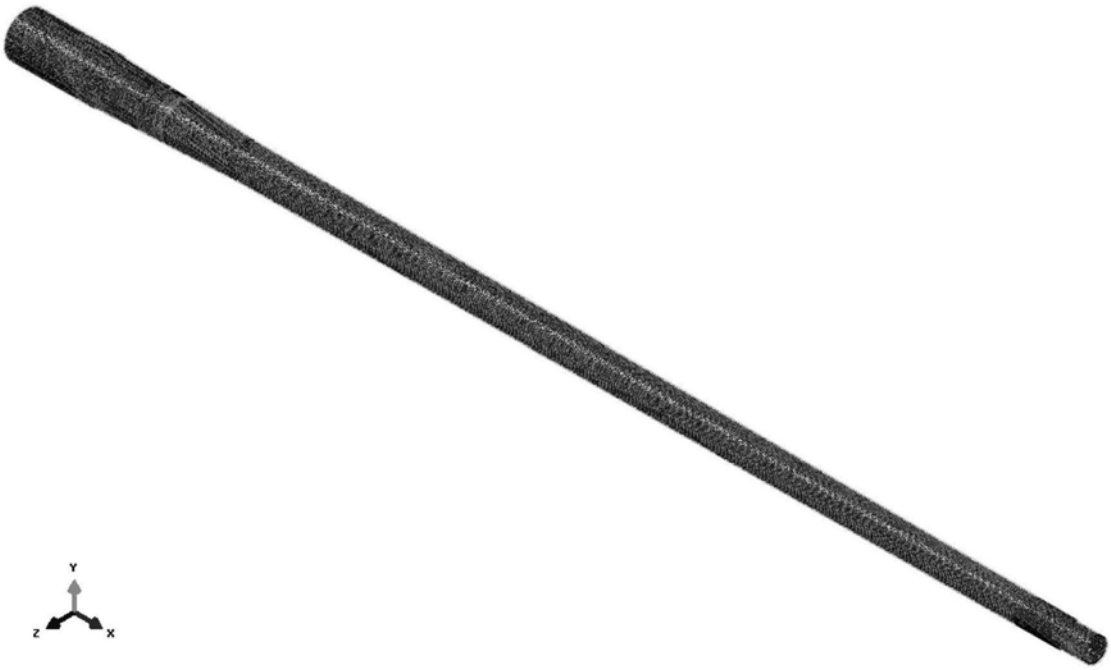


图11

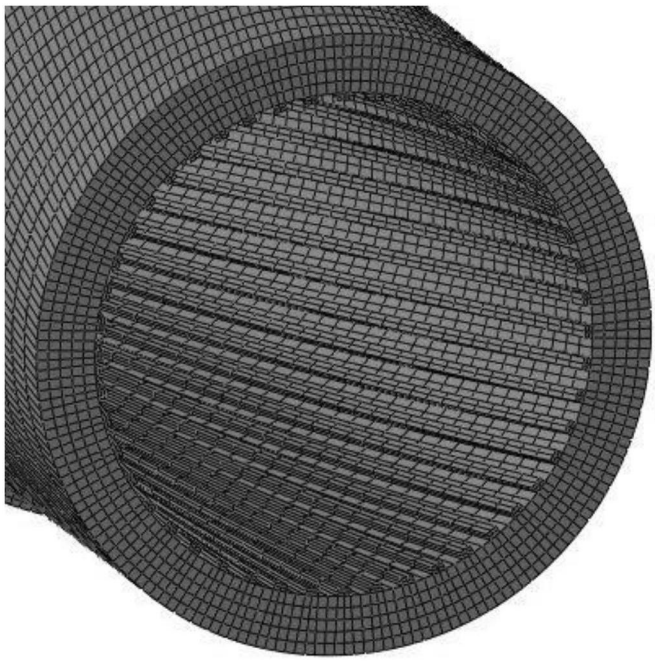


图12

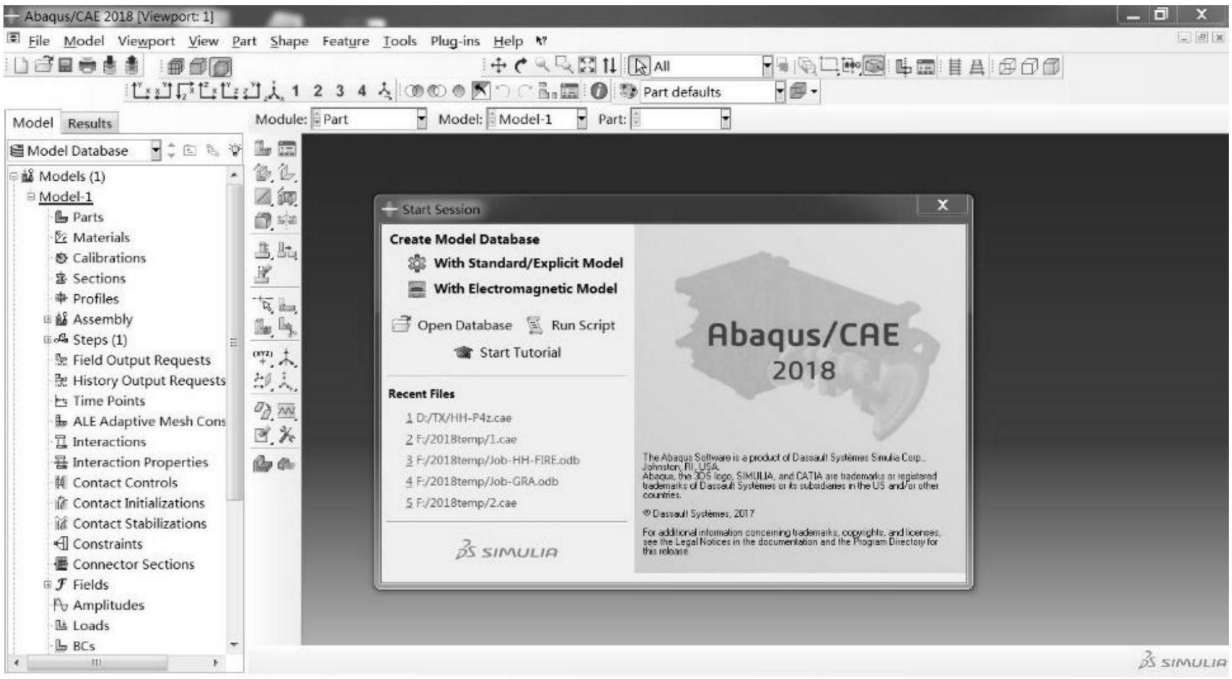

Abaqus/CAE 2018 [Viewport: 1]
File Model Viewport View Part Shape Feature Tools Plug-ins Help
Model Results
Model Database
Models (1)
Model-1
Parts
Materials
Calibrations
Sections
Profiles
Assembly
Steps (1)
Field Output Requests
History Output Requests
Time Points
ALE Adaptive Mesh Cons
Interactions
Interaction Properties
Contact Controls
Contact Initializations
Contact Stabilizations
Constraints
Connector Sections
Fields
Amplitudes
Loads
BCs
Module: Part
Model: Model-1
Part:
Start Session
Create Model Database
With Standard/Explicit Model
With Electromagnetic Model
Open Database
Run Script
Start Tutorial
Recent Files
1 D:/TX/HH-P4z.cae
2 F:/2018temp/1.cae
3 F:/2018temp/Job-HH-FIRE.odb
4 F:/2018temp/Job-GRA.odb
5 F:/2018temp/2.cae
SIMULIA
Abaqus/CAE
2018
SIMULIA

图13

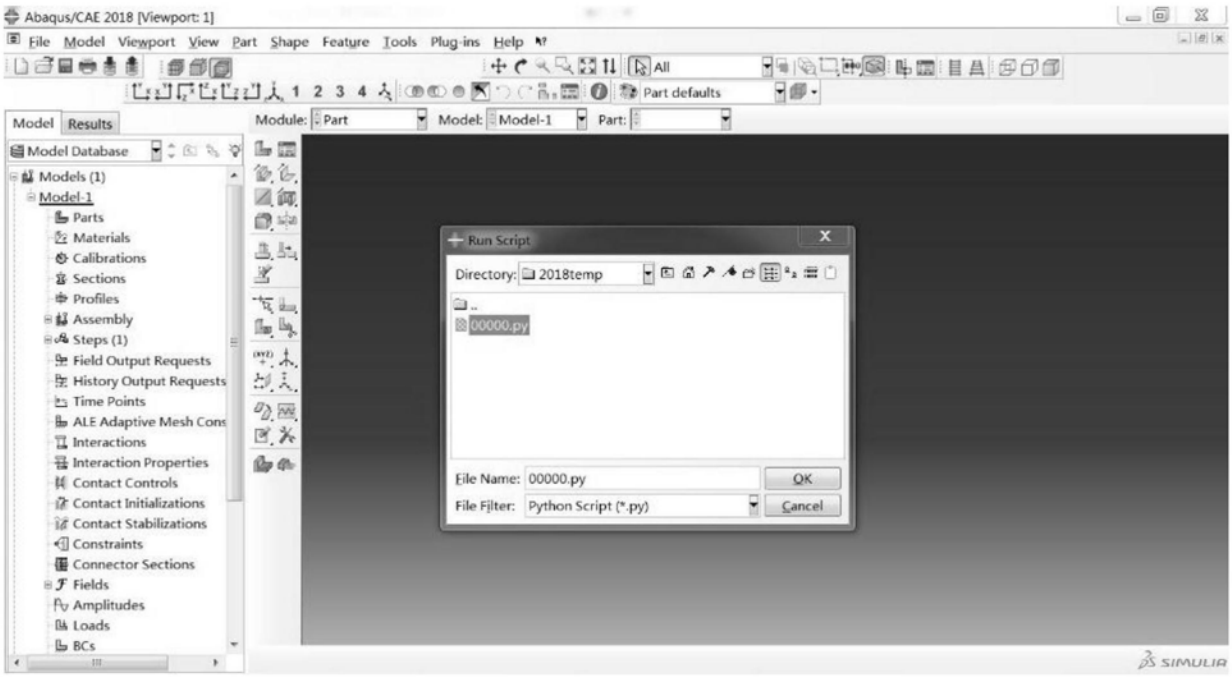

Abaqus/CAE 2018 [Viewport: 1]
File Model Viewport View Part Shape Feature Tools Plug-ins Help
Model Results
Model Database
Models (1)
Model-1
Parts
Materials
Calibrations
Sections
Profiles
Assembly
Steps (1)
Field Output Requests
History Output Requests
Time Points
ALE Adaptive Mesh Cons
Interactions
Interaction Properties
Contact Controls
Contact Initializations
Contact Stabilizations
Constraints
Connector Sections
Fields
Amplitudes
Loads
BCs
Module: Part
Model: Model-1
Part:
Run Script
Directory: 2018temp
00000.py
File Name: 00000.py
OK
File Filter: Python Script (*.py)
Cancel
SIMULIA

图14

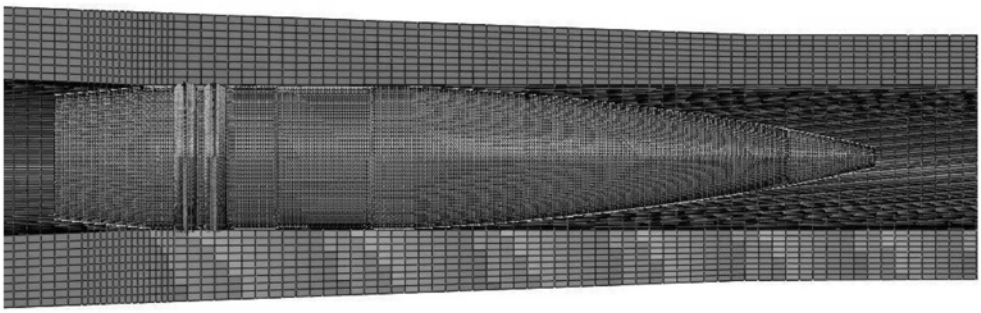


图15